



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/081,682	02/25/2002	Amnon Naamad	07072-946001	2103
26161	7590	08/01/2006	EXAMINER	
FISH & RICHARDSON PC P.O. BOX 1022 MINNEAPOLIS, MN 55440-1022			IWASHKO, LEV	
			ART UNIT	PAPER NUMBER
			2186	

DATE MAILED: 08/01/2006

Please find below and/or attached an Office communication concerning this application or proceeding..

## Office Action Summary

### Application No.

10/081,682

### Applicant(s)

NAAMAD ET AL.

### Examiner

Lev I. Iwashko

### Art Unit

2186

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 05 June 2002.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-29 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-29 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 20 February 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date 2/20/2003.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_.

## DETAILED ACTION

### *Response to Amendment*

1. There were no cancelled or amended claims in the Applicant's arguments.
2. Claims 1-29 stand rejected.

### *Claim Rejections - 35 USC § 102*

3. The following are quotations of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1-9, 11-21, and 23-28 are rejected under U.S.C. 102(b) as being anticipated by Candelaria et al. (US Patent 5,815,656).

Claim 1. In a data-storage system having a data storage unit that includes at least two constituent data storage elements, (*Column 8, lines 59-63 – State the following: “In some systems redundant records exist in an out of synchronization cylinder range. Existence of such a range is verified at step 148. If all the tracks test is valid step 150 we may de-stage the tracks from the out of synchronization cylinder range (step 152).” The above lines simply state that there is a cylinder-data-storage system that includes “tracks” which denote at least 2 constituent storage elements.)*

- each of said constituent data storage elements being in one of a first state and a second state other than said first state, a method comprising: (*Column 8, line – States “At step 112 the nonvolatile storage is set to the failed state.” Column 8, lines 41-42 – State “If YES, nonvolatile storage is set to the unavailable state.” The above two sentences show that there are at least 2 states for the storage elements)*

Art Unit: 2186

- providing a data structure having an entry corresponding to said data storage unit, said entry including status information indicating whether at least one constituent data storage element of said data storage unit is in said first state; and *(Column 7, lines 16-27 – State the following: “The directory entry is a control block which indicates location of a track image in cache and the address in DASD of the unmodified track. If a track image is the subject of a channel command word chain, it is marked busy. An additional data structure is referred to as the track information block (TIB). The track information block has addresses for each record of a track that is in the cache. In the first preferred embodiment, a process accessing records listed in the TIB adds information relating to modification status of the record images. If a record image has been modified, but not de-staged to DASD, it is listed as a modified record in the TIB”. The above sentences show an entry, as well as 2 possible states that they could be in)*
- updating said entry following a change in state of at least one of said constituent data storage elements. *(Column 7, lines 31-32 – State the following: “An update write modifies a record existing on a DASD”)*

Claim 2. The method of claim 1, wherein updating said entry comprises:

- identifying an entry in said data structure corresponding to a data storage unit that includes a constituent data storage element in said first state; *(Column 7, lines 22-25 – State the following: “In the first preferred embodiment, a process accessing records listed in the TIB adds information relating to modification status of the record images”)*
- modifying status information in said entry to indicate that said data storage unit includes at least one constituent data storage element in said first state. *(Column 7, lines 32-38 – State the following: “A format write contains new records. Upon receipt of an Update chain, all records of a track image written by the channel program are*

*scheduled modified as they are written. Modification is marked in the track information block which is in a special store owned by the storage path. The track information block is moved to cache when the operation completes”)*

Claim 3. The method of claim 2, further comprising locking said data structure before modifying status information and unlocking said data structure after modifying status information. *(Column 9, lines 7-46 – State the following: “Cache data structure congruency is enforced by cache locks. Cache locks serialize access to cache control structures among the storage paths. Only one storage path can own a lock at a given time. Upon recovery, the more conservative approach would assume that a control structure is incongruent if it was owned by a storage path when the storage path failed. However, greater speed in recovery is obtained by further refinement of the process. The locks are granted to the following resources: (1) the scatter index table, used to index directory entries; (2) the pools of available directory entries; (3) the lists of active directory entries (i.e. the MRU/LRU lists); (4) a hash table used to index locked shared data; and (5) the pool of prepared messages waiting to be off loaded to a host that describe the status of the subject to host user defined locks shared data. Locks are granted to items 3 and 4 with particular frequently. The data structure used to support validation of user records is a congruency matrix which is maintained in the shared control arrays 64 and 66 by a background process illustrated in FIG. 6. The process is entered at step 200 which is a repeating test used to determine if a lock has been granted to a storage path for a resource. Incongruence does not occur absent existence of such a lock. After issuance of such a lock, step 202 is executed to mark the resources as locked in the congruency matrix. However, a definition of incongruence equated with existence of a lock is often overboard. Secondary marks can be used to define situations actually associated with data consistency far more closely than a*

*"resource lock." The concept of incongruence may also be extended to cover portions of code responding to a channel command word chain that creates fast write data in cache and the nonvolatile storage. Write operations to a single linked free list are set by a resource lock to include a whole period between receipt and release of the lock. However, the list becomes incongruent only during an actual write operation. For maximum performance, incongruence is signaled by setting secondary marks immediately prior to the write and resetting the mark directly upon completion of the write. The operation is set forth by process steps 204 (following the NO branch), 206, 208, 210 and 212")*

Claim 4. The method of claim 2, wherein modifying status information comprises inspecting said status information to determine if said status information already indicates that at least one constituent data storage element is in said first state. (Column 9, lines 47-65 – State the following: *"Consider a channel command chain which is a write hit to a previously unmodified record. Further suppose that the channel end and device end signals have been presented to the host. The cache is in an incongruent state with a DASD record until the modified record is de-staged. Such incongruence is marked in the matrix in the same fashion as the resource lock, but at a secondary level. Step 204 is used to determine if a channel command chain may include such a write operation. If a secondary mark set is used, the YES branch advances the process to step 206 where it is determined if the conditions required for setting a secondary mark in the congruency matrix exist. If the conditions do exist, the mark is set in step 208 and the operation is monitored until completed (step 210). With execution of step 212, the secondary mark is reset to indicate that incongruence no longer exists. If the NO branch from step 206 was taken no secondary level mark of incongruence is ever set. Such absence of activity would be associated with a read operation"*)

Claim 5. The method of claim 1, wherein updating said entry comprises:

- detecting that a constituent data storage element is in said second state; *(Column 7, lines 35-37 – State the following: “Modification is marked in the track information block which is in a special store owned by the storage path”)*
- determining whether said data storage unit contains any constituent data storage element in said first state; *(Column 7, lines 17-25 – State the following: “If a track image is the subject of a channel command word chain, it is marked busy. An additional data structure is referred to as the track information block (TIB). The track information block has addresses for each record of a track that is in the cache. In the first preferred embodiment, a process accessing records listed in the TIB adds information relating to modification status of the record images”)*
- identifying an entry in said data structure corresponding to a data storage unit that includes said constituent data storage element; *(Column 7, lines 22-25 – State the following: “In the first preferred embodiment, a process accessing records listed in the TIB adds information relating to modification status of the record images”)*
- modifying status information in said entry to indicate that no constituent data storage elements of said data storage unit are in said first state. *(Column 7, lines 32-38 – State the following: “A format write contains new records. Upon receipt of an Update chain, all records of a track image written by the channel program are scheduled modified as they are written. Modification is marked in the track information block which is in a special store owned by the storage path. The track information block is moved to cache when the operation completes”)*

Claim 6. The method of claim 5, further comprising locking said data structure before modifying status information and unlocking said data structure after modifying status information. *(Column 9, lines 7-46 – State the*

*following: "Cache data structure congruency is enforced by cache locks. Cache locks serialize access to cache control structures among the storage paths. Only one storage path can own a lock at a given time. Upon recovery, the more conservative approach would assume that a control structure is incongruent if it was owned by a storage path when the storage path failed. However, greater speed in recovery is obtained by further refinement of the process. The locks are granted to the following resources: (1) the scatter index table, used to index directory entries; (2) the pools of available directory entries; (3) the lists of active directory entries (i.e. the MRU/LRU lists); (4) a hash table used to index locked shared data; and (5) the pool of prepared messages waiting to be off loaded to a host that describe the status of the subject to host user defined locks shared data. Locks are granted to items 3 and 4 with particular frequently. The data structure used to support validation of user records is a congruency matrix which is maintained in the shared control arrays 64 and 66 by a background process illustrated in FIG. 6. The process is entered at step 200 which is a repeating test used to determine if a lock has been granted to a storage path for a resource. Incongruence does not occur absent existence of such a lock. After issuance of such a lock, step 202 is executed to mark the resources as locked in the congruency matrix. However, a definition of incongruence equated with existence of a lock is often overboard. Secondary marks can be used to define situations actually associated with data consistency far more closely than a "resource lock." The concept of incongruence may also be extended to cover portions of code responding to a channel command word chain that creates fast write data in cache and the nonvolatile storage. Write operations to a single linked free list are set by a resource lock to include a whole period between receipt and release of the lock. However, the list becomes incongruent only during an actual write operation. For maximum performance, incongruence is signaled by setting secondary*



*marks immediately prior to the write and resetting the mark directly upon completion of the write. The operation is set forth by process steps 204 (following the NO branch), 206, 208, 210 and 212”)*

Claim 7. The method of claim 5, wherein modifying status information comprises inspecting said status information to determine if said status information already indicates that all constituent data storage elements are in said second state. *(Column 9, lines 47-65 – State the following: “Consider a channel command chain which is a write hit to a previously unmodified record. Further suppose that the channel end and device end signals have been presented to the host. The cache is in an incongruent state with a DASD record until the modified record is de-staged. Such incongruence is marked in the matrix in the same fashion as the resource lock, but at a secondary level. Step 204 is used to determine if a channel command chain may include such a write operation. If a secondary mark set is used, the YES branch advances the process to step 206 where it is determined if the conditions required for setting a secondary mark in the congruency matrix exist. If the conditions do exist, the mark is set in step 208 and the operation is monitored until completed (step 210). With execution of step 212, the secondary mark is reset to indicate that incongruence no longer exists. If the NO branch from step 206 was taken no secondary level mark of incongruence is ever set. Such absence of activity would be associated with a read operation”)*

Claim 8. The method of claim 1, further comprising selecting said data storage unit to be a cylinder and selecting said constituent data storage elements to be tracks included in said cylinder. *(Column 8, lines 59-63 – State the following: “In some systems redundant records exist in an out of synchronization cylinder range. Existence of such a range is verified at step 148. If all the tracks test is valid step 150 we may de-stage the tracks from the out of synchronization cylinder range (step 152).” The above*

*lines simply state that there is a cylinder-data-storage system that includes "tracks" which denote at least 2 constituent storage elements.)*

- Claim 9. The method of claim 8 her comprising selecting said first state to indicate the presence of invalid data on said track. *(Column 8, lines 7-34 – State the following: "At step 112 the nonvolatile storage is set to the failed state. Step 114 controls looping through the directory entry validation process. Step 114 determines if all directory entries have been checked. Once all entries have been checked the process follows the YES branch to off page connector "A". The NO branch is followed until all directory entries have been examined. Along the NO branch step 116 is executed, which attempts access the next directory entry. Because all internal users have freed the cache no directory entry should be busy. At step 118 it is determined if the directory entry is busy. If the directory is busy, error has occurred. If a directory entry is busy the process is exited along the YES branch from step 118 through off page connector B. The NO branch of from step 118 indicates continued evaluation of the integrity of the data structures. Along the NO branch to step 120 we begin use of the scatter index table to validate cache structures. The hash chains of the scatter index table are traversed to verify that no chains have been broken (steps 120 and 122). If no chains are broken, the tracks information block is read and compared (step 124) against directory entries. If there are incongruencies in the data of the tracks information block in the directory entries, the process is exited by off page connector B. If there are no incongruencies, step 128 is executed to add a status indication to one of the active, pinned or defective lists for tracks. If a segment is unused, the segment is added to a free list at step 130 and the process is returned to step 114 to determine if directory entries remain for evaluation")*
- Claim 11. The method of claim 1, further comprising scanning said data structure to locate constituent data storage elements in said first state. *(Column 7, lines*

*16-18 – State the following, which shows how the directory cache “scans” the cache for tracks in an unmodified state)*

- Claim 12. The method of claim 11, wherein scanning said data structure comprises:
- detecting an entry in said data structure that indicates the presence, in said data storage unit associated with said data structure, of at least one constituent data storage element in said first state; *(Column 7, lines 22-25 – State the following: “In the first preferred embodiment, a process accessing records listed in the TIB adds information relating to modification status of the record images”)*
  - and scanning constituent data storage elements included in said data storage unit to identify said constituent data storage element in said first state. *(Column 7, lines 32-38 – State the following: “A format write contains new records. Upon receipt of an Update chain, all records of a track image written by the channel program are scheduled modified as they are written. Modification is marked in the track information block which is in a special store owned by the storage path. The track information block is moved to cache when the operation completes”)*
- Claim 13. A computer-readable medium having software for execution in *(Column 7, lines 34 – Mentions a “channel program”, which inherently infers a software)*
- a data-storage system having a data storage unit that includes at least two constituent data storage elements, *(Column 8, lines 59-63 – State the following: “In some systems redundant records exist in an out of synchronization cylinder range. Existence of such a range is verified at step 148. If all the tracks test is valid step 150 we may de-stage the tracks from the out of synchronization cylinder range (step 152).” The above lines simply state that there is a cylinder-data-storage system that includes “tracks” which denote at least 2 constituent storage elements.)*

Art Unit: 2186

- each of said constituent data storage elements being in one of a first state and a second state other than said first state, said software comprising instructions for: *(Column 8, line – States “At step 112 the nonvolatile storage is set to the failed state.” Column 8, lines 41-42 – State “If YES, nonvolatile storage is set to the unavailable state.” The above two sentences show that there are at least 2 states for the storage elements)*
- providing a data structure having an entry corresponding to said data storage unit, said entry including status information indicating whether at least one constituent data storage element of said data storage unit is in said first state; *(Column 7, lines 16-27 – State the following: “The directory entry is a control block which indicates location of a track image in cache and the address in DASD of the unmodified track. If a track image is the subject of a channel command word chain, it is marked busy. An additional data structure is referred to as the track information block (TIB). The track information block has addresses for each record of a track that is in the cache. In the first preferred embodiment, a process accessing records listed in the TIB adds information relating to modification status of the record images. If a record image has been modified, but not de-staged to DASD, it is listed as a modified record in the TIB”. The above sentences show an entry, as well as 2 possible states that they could be in)*
- and updating said entry following a change in state of at least one of said constituent data storage elements. *(Column 7, lines 31-32 – State the following: “An update write modifies a record existing on a DASD”)*

Claim 14. The computer-readable medium of claim 13, wherein said instructions for updating said entry comprise instructions for:

- identifying an entry in said data structure corresponding to a data storage unit that includes a constituent data storage element in said

first state; (Column 7, lines 22-25 – State the following: “In the first preferred embodiment, a process accessing records listed in the TIB adds information relating to modification status of the record images”)

- modifying status information in said entry to indicate that said data storage unit includes at least one constituent data storage element in said first state. (Column 7, lines 32-38 – State the following: “A format write contains new records. Upon receipt of an Update chain, all records of a track image written by the channel program are scheduled modified as they are written. Modification is marked in the track information block which is in a special store owned by the storage path. The track information block is moved to cache when the operation completes”)

Claim 15. The computer-readable medium of claim 14, wherein said software further comprises instructions for locking said data structure before modifying status information and unlocking said data structure after modifying status information. (Column 9, lines 7-46 – State the following: “Cache data structure congruency is enforced by cache locks. Cache locks serialize access to cache control structures among the storage paths. Only one storage path can own a lock at a given time. Upon recovery, the more conservative approach would assume that a control structure is incongruent if it was owned by a storage path when the storage path failed. However, greater speed in recovery is obtained by further refinement of the process. The locks are granted to the following resources: (1) the scatter index table, used to index directory entries; (2) the pools of available directory entries; (3) the lists of active directory entries (i.e. the MRU/LRU lists); (4) a hash table used to index locked shared data; and (5) the pool of prepared messages waiting to be off loaded to a host that describe the status of the subject to host user defined locks shared data. Locks are granted to items 3 and 4 with particular

*frequently. The data structure used to support validation of user records is a congruency matrix which is maintained in the shared control arrays 64 and 66 by a background process illustrated in FIG. 6. The process is entered at step 200 which is a repeating test used to determine if a lock has been granted to a storage path for a resource. Incongruence does not occur absent existence of such a lock. After issuance of such a lock, step 202 is executed to mark the resources as locked in the congruency matrix. However, a definition of incongruence equated with existence of a lock is often overboard. Secondary marks can be used to define situations actually associated with data consistency far more closely than a "resource lock." The concept of incongruence may also be extended to cover portions of code responding to a channel command word chain that creates fast write data in cache and the nonvolatile storage. Write operations to a single linked free list are set by a resource lock to include a whole period between receipt and release of the lock. However, the list becomes incongruent only during an actual write operation. For maximum performance, incongruence is signaled by setting secondary marks immediately prior to the write and resetting the mark directly upon completion of the write. The operation is set forth by process steps 204 (following the NO branch), 206, 208, 210 and 212")*

Claim 16. The computer-readable medium of claim 14, wherein said instructions for modifying status information comprise instructions for inspecting said status information to determine if said status information already indicates that at least one constituent data storage element is in said first state. (Column 9, lines 47-65 – State the following: “Consider a channel command chain which is a write hit to a previously unmodified record. Further suppose that the channel end and device end signals have been presented to the host. The cache is in an incongruent state with a DASD record until the modified record is de-staged. Such incongruence is marked in the matrix in the same fashion as the resource lock, but at a

*secondary level. Step 204 is used to determine if a channel command chain may include such a write operation. If a secondary mark set is used, the YES branch advances the process to step 206 where it is determined if the conditions required for setting a secondary mark in the congruency matrix exist. If the conditions do exist, the mark is set in step 208 and the operation is monitored until completed (step 210). With execution of step 212, the secondary mark is reset to indicate that incongruence no longer exists. If the NO branch from step 206 was taken no secondary level mark of incongruence is ever set. Such absence of activity would be associated with a read operation”)*

- Claim 17. The computer-readable medium of claim 13, wherein said instructions for updating said entry comprise instructions for:
- detecting that a constituent data storage element is in said second state; *(Column 7, lines 35-37 – State the following: “Modification is marked in the track information block which is in a special store owned by the storage path”)*
  - determining whether said data storage unit contains any constituent data storage element in said first state; *(Column 7, lines 17-25 – State the following: “If a track image is the subject of a channel command word chain, it is marked busy. An additional data structure is referred to as the track information block (TIB). The track information block has addresses for each record of a track that is in the cache. In the first preferred embodiment, a process accessing records listed in the TIB adds information relating to modification status of the record images”)*
  - identifying an entry in said data structure corresponding to a data storage unit that includes said constituent data storage element; *(Column 7, lines 22-25 – State the following: “In the first preferred embodiment, a process accessing records listed in the TIB adds information relating to modification status of the record images”)*

Art Unit: 2186

- modifying status information in said entry to indicate that no constituent data storage elements of said data storage unit are in said first state. *(Column 7, lines 32-38 – State the following: “A format write contains new records. Upon receipt of an Update chain, all records of a track image written by the channel program are scheduled modified as they are written. Modification is marked in the track information block which is in a special store owned by the storage path. The track information block is moved to cache when the operation completes”)*

Claim 18. The computer-readable medium of claim 17, wherein said software comprises instructions for locking said data structure before modifying status information and unlocking said data structure after modifying status information. *(Column 9, lines 7-46 – State the following: “Cache data structure congruency is enforced by cache locks. Cache locks serialize access to cache control structures among the storage paths. Only one storage path can own a lock at a given time. Upon recovery, the more conservative approach would assume that a control structure is incongruent if it was owned by a storage path when the storage path failed. However, greater speed in recovery is obtained by further refinement of the process. The locks are granted to the following resources: (1) the scatter index table, used to index directory entries; (2) the pools of available directory entries; (3) the lists of active directory entries (i.e. the MRU/LRU lists); (4) a hash table used to index locked shared data; and (5) the pool of prepared messages waiting to be off loaded to a host that describe the status of the subject to host user defined locks shared data. Locks are granted to items 3 and 4 with particular frequently. The data structure used to support validation of user records is a congruency matrix which is maintained in the shared control arrays 64 and 66 by a background process illustrated in FIG. 6. The process is entered at step 200 which is a repeating test used to determine if a lock*



*has been granted to a storage path for a resource. Incongruence does not occur absent existence of such a lock. After issuance of such a lock, step 202 is executed to mark the resources as locked in the congruency matrix. However, a definition of incongruence equated with existence of a lock is often overboard. Secondary marks can be used to define situations actually associated with data consistency far more closely than a "resource lock." The concept of incongruence may also be extended to cover portions of code responding to a channel command word chain that creates fast write data in cache and the nonvolatile storage. Write operations to a single linked free list are set by a resource lock to include a whole period between receipt and release of the lock. However, the list becomes incongruent only during an actual write operation. For maximum performance, incongruence is signaled by setting secondary marks immediately prior to the write and resetting the mark directly upon completion of the write. The operation is set forth by process steps 204 (following the NO branch), 206, 208, 210 and 212")*

- Claim 19. The computer-readable medium of claim 17, wherein said instructions for modifying status information comprise instructions for inspecting said status information to determine if said status information already indicates that all constituent data storage elements are in said second state. (Column 9, lines 47-65 – State the following: “Consider a channel command chain which is a write hit to a previously unmodified record. Further suppose that the channel end and device end signals have been presented to the host. The cache is in an incongruent state with a DASD record until the modified record is de-staged. Such incongruence is marked in the matrix in the same fashion as the resource lock, but at a secondary level. Step 204 is used to determine if a channel command chain may include such a write operation. If a secondary mark set is used, the YES branch advances the process to step 206 where it is determined if the conditions required for setting a secondary mark in the congruency matrix exist. If

*the conditions do exist, the mark is set in step 208 and the operation is monitored until completed (step 210). With execution of step 212, the secondary mark is reset to indicate that incongruence no longer exists. If the NO branch from step 206 was taken no secondary level mark of incongruence is ever set. Such absence of activity would be associated with a read operation")*

Claim 20. The computer-readable medium of claim 13, wherein said software her comprises instructions for selecting said data storage unit to be a cylinder and selecting said constituent data storage elements to be tracks included in said cylinder. *(Column 8, lines 59-63 – State the following: “In some systems redundant records exist in an out of synchronization cylinder range. Existence of such a range is verified at step 148. If all the tracks test is valid step 150 we may de-stage the tracks from the out of synchronization cylinder range (step 152).” The above lines simply state that there is a cylinder-data-storage system that includes “tracks” which denote at least 2 constituent storage elements.)*

Claim 21. The computer-readable medium of claim 8, wherein said software further comprises instructions for selecting said first state to indicate the presence of invalid data on said track. *(Column 8, lines 7-34 – State the following: “At step 112 the nonvolatile storage is set to the failed state. Step 114 controls looping through the directory entry validation process. Step 114 determines if all directory entries have been checked. Once all entries have been checked the process follows the YES branch to off page connector “A”. The NO branch is followed until all directory entries have been examined. Along the NO branch step 116 is executed, which attempts access the next directory entry. Because all internal users have freed the cache no directory entry should be busy. At step 118 it is determined if the directory entry is busy. If the directory is busy, error has occurred. If a directory entry is busy the process is exited along the YES branch from step 118 through off page connector B. The NO branch of*

*from step 118 indicates continued evaluation of the integrity of the data structures. Along the NO branch to step 120 we begin use of the scatter index table to validate cache structures. The hash chains of the scatter index table are traversed to verify that no chains have been broken (steps 120 and 122). If no chains are broken, the tracks information block is read and compared (step 124) against directory entries. If there are incongruencies in the data of the tracks information block in the directory entries, the process is exited by off page connector B. If there are no incongruencies, step 128 is executed to add a status indication to one of the active, pinned or defective lists for tracks. If a segment is unused, the segment is added to a free list at step 130 and the process is returned to step 114 to determine if directory entries remain for evaluation")*

Claim 23. The computer-readable medium of claim 13, wherein said software further comprises instructions for scanning said data structure to locate constituent data storage elements in said first state. *(Column 7, lines 16-18 – State the following, which shows how the directory cache “scans” the cache for tracks in an unmodified state)*

Claim 24. The computer-readable medium of claim 11, wherein said instructions for scanning said data structure comprise instructions for:

- detecting an entry in said data structure that indicates the presence, in said data storage unit associated with said data structure, of at least one constituent data storage element in said first state; *(Column 7, lines 22-25 – State the following: “In the first preferred embodiment, a process accessing records listed in the TIB adds information relating to modification status of the record images”)*
- and scanning constituent data storage elements included in said data storage unit to identify said constituent data storage element in said first state. *(Column 7, lines 32-38 – State the following: “A format write contains new records. Upon receipt of an Update chain, all records of a track image written by the channel program are*

*scheduled modified as they are written. Modification is marked in the track information block which is in a special store owned by the storage path. The track information block is moved to cache when the operation completes”)*

Claim 25. A data-storage system comprising:

- a data storage unit that includes at least two constituent data storage elements, *(Column 8, lines 59-63 – State the following: “In some systems redundant records exist in an out of synchronization cylinder range. Existence of such a range is verified at step 148. If all the tracks test is valid step 150 we may de-stage the tracks from the out of synchronization cylinder range (step 152).” The above lines simply state that there is a cylinder-data-storage system that includes “tracks” which denote at least 2 constituent storage elements.)*
- each of said constituent data storage elements being in one of a first state and a second state other than said first state: *(Column 8, line – States “At step 112 the nonvolatile storage is set to the failed state.” Column 8, lines 41-42 – State “If YES, nonvolatile storage is set to the unavailable state.” The above two sentences show that there are at least 2 states for the storage elements)*
- a memory element configured to hold a data structure having an entry corresponding to said data storage unit, said entry including status information indicating whether at least one constituent data storage element of said data storage unit is in said first state.

Claim 26. The data-storage system of claim 25, further comprising a lock for locking

said data structure to prevent modification of said status information.

*(Column 9, lines 7-46 – State the following: “Cache data structure congruency is enforced by cache locks. Cache locks serialize access to cache control structures among the storage paths. Only one storage path can own a lock at a given time. Upon recovery, the more conservative approach would assume that a control structure is incongruent if it was*

*owned by a storage path when the storage path failed. However, greater speed in recovery is obtained by further refinement of the process. The locks are granted to the following resources: (1) the scatter index table, used to index directory entries; (2) the pools of available directory entries; (3) the lists of active directory entries (i.e. the MRU/LRU lists); (4) a hash table used to index locked shared data; and (5) the pool of prepared messages waiting to be off loaded to a host that describe the status of the subject to host user defined locks shared data. Locks are granted to items 3 and 4 with particular frequently. The data structure used to support validation of user records is a congruency matrix which is maintained in the shared control arrays 64 and 66 by a background process illustrated in FIG. 6. The process is entered at step 200 which is a repeating test used to determine if a lock has been granted to a storage path for a resource. Incongruence does not occur absent existence of such a lock. After issuance of such a lock, step 202 is executed to mark the resources as locked in the congruency matrix. However, a definition of incongruence equated with existence of a lock is often overboard. Secondary marks can be used to define situations actually associated with data consistency far more closely than a "resource lock." The concept of incongruence may also be extended to cover portions of code responding to a channel command word chain that creates fast write data in cache and the nonvolatile storage. Write operations to a single linked free list are set by a resource lock to include a whole period between receipt and release of the lock. However, the list becomes incongruent only during an actual write operation. For maximum performance, incongruence is signaled by setting secondary marks immediately prior to the write and resetting the mark directly upon completion of the write. The operation is set forth by process steps 204 (following the NO branch), 206, 208, 210 and 212")*

Art Unit: 2186

- Claim 27. The data-storage system of claim 25, wherein said data storage unit comprises a cylinder and said constituent data storage elements comprise tracks included in said cylinder. *(Column 8, lines 59-63 – State the following: “In some systems redundant records exist in an out of synchronization cylinder range. Existence of such a range is verified at step 148. If all the tracks test is valid step 150 we may de-stage the tracks from the out of synchronization cylinder range (step 152).” The above lines simply state that there is a cylinder-data-storage system that includes “tracks” which denote at least 2 constituent storage elements.)*
- Claim 28. The data-storage system of claim 27, wherein said first state indicates the presence of invalid data on said track. *(Column 8, lines 7-34 – State the following: “At step 112 the nonvolatile storage is set to the failed state. Step 114 controls looping through the directory entry validation process. Step 114 determines if all directory entries have been checked. Once all entries have been checked the process follows the YES branch to off page connector “A”. The NO branch is followed until all directory entries have been examined. Along the NO branch step 116 is executed, which attempts access the next directory entry. Because all internal users have freed the cache no directory entry should be busy. At step 118 it is determined if the directory entry is busy. If the directory is busy, error has occurred. If a directory entry is busy the process is exited along the YES branch from step 118 through off page connector B. The NO branch of from step 118 indicates continued evaluation of the integrity of the data structures. Along the NO branch to step 120 we begin use of the scatter index table to validate cache structures. The hash chains of the scatter index table are traversed to verify that no chains have been broken (steps 120 and 122). If no chains are broken, the tracks information block is read and compared (step 124) against directory entries. If there are incongruencies in the data of the tracks information block in the directory entries, the process is exited by off page connector B. If there are no*

*incongruencies, step 128 is executed to add a status indication to one of the active, pinned or defective lists for tracks. If a segment is unused, the segment is added to a free list at step 130 and the process is returned to step 114 to determine if directory entries remain for evaluation”)*

***Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

- (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 10, 22, and 29 are rejected under 35 U.S.C.103(a) as being unpatentable over Candelaria et al. as applied to claims 1, 8, 25, and 27 above, further in view of Mikkelsen et al. (US Patent 6539462 B1).

Candelaria teaches the limitations of claims 1, 8, 25, and 27 for the reasons above.

Candelaria’s invention differs from the claimed invention in that there is no specific reference to a bit map and bits corresponding to the cylinder.

Candelaria fails to teach claims 10, 22, and 29, which respectively state: “The method of claim 8, wherein providing a data structure comprises providing a map having a plurality of bits, each of which corresponds to a cylinder, each bit having a first state indicating that at least one track in said cylinder includes invalid data and a second state indicating that no tracks in said cylinder include invalid data”, “The computer-readable medium of claim 8, wherein said instructions for providing a data structure comprise instructions for providing a bit map having a plurality of bits, each of which corresponds to a cylinder, each bit having a first state indicating

Art Unit: 2186

that at least one track in said cylinder includes invalid data and a second state indicating that no tracks in said cylinder include invalid data”, and “The data-storage system of claim 27, wherein said data structure comprises a bit map having a plurality of bits, each of which corresponds to a cylinder, each bit having a first state indicating that at least one track in said cylinder includes invalid data and a second state indicating that no tracks in said cylinder include invalid data.” However, Mikkelsen claims “In the embodiments discussed above, this may be done by examining the bit map to determine if any bits are set, indicating which track or cylinder contains updated information that is not included in the remote copy on secondary data recording device DASD-2.” (Column 9, lines 60-65). It would have been obvious to one of ordinary skill in the art, having the teachings of the “Method of Validating Memory Structures in Data Processing Systems” of Candelaria and Mikkelsen’s “Remote data copy using a prospective suspend command” before him at the time the invention was made, to combine the inventions to include a bit map and bits corresponding to the cylinder, so that the system would run more accurately and efficiently.



*Response to Arguments*

6. Applicant's arguments (filed June 5, 2006) with respect to claims 1-29 have been fully considered but are not persuasive.

7. With regards to Claim 1, the Applicant alleges "Candelaria fails to teach 'providing a data structure'". However, if Claim 1 reads "an entry corresponding to said data storage unit", and the proposed "data storage element" of Candelaria falls within the unit, then Candelaria successfully teaches "providing a data structure. Therefore, the Applicant's arguments are moot over the prior art.

8. Further regarding Claim 1, the Applicant alleges that "Candelaria fails to teach 'updating said entry'". However, Candelaria discloses the following: "Where a track image is present in cache, a write command from a host computer updating a record of the track results in the record being written over the corresponding record in the track image rather than to DASD" (Column 3, lines 34-37). Therefore, since the entry is shown to have a potential update occur, the Applicant's arguments are moor in view of the prior art.

9. Claims 2-12 remain rejected due to their dependence on rejected independent Claim 1.

10. Claims 13-25 remain rejected due to no new arguments being presented in their defense.

11. With regard to Claim 8 and 20, the Applicant alleges that the mapping of the claim elements fail in Candelaria. The Examiner maintains that the mappings do in fact coincide, as explained in Claim 1. Therefore, the Applicant's arguments remain moot over the prior art.

12. With regard to Claims 11 and 23, the Applicant alleges that the "disclosure of scanning a cache is not a disclosure of scanning said data structure". However, Candelaria discloses the following: "If no chains are broken, the track information block is read and compared (step 124)

Art Unit: 2186

against directory entries” (Column 8, lines 25-27). This denotes that the TIB (data structure is scanned, and therefore the Applicant’s arguments are moot in view of the prior art.

***Conclusion***

13. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lev I. Iwashko whose telephone number is (571)272-1658. The examiner can normally be reached on M-Th, from 8-6PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Matt Kim can be reached on (571)272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Art Unit: 2186

Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Lev Iwashko



MATTHEW KIM  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100